Bitcoins y el problema de los generales bizantinos

Cristina Pérez-Solà
Departament d'enginyeria de la informació i les comunicacions
Universitat Autònoma de Barcelona
Email: cperez@deic.uab.cat

Jordi Herrera-Joancomartí
Departament d'enginyeria de la informació i les comunicacions
Universitat Autònoma de Barcelona
Email: jordi.herrera@uab.cat

***Resumen*—En este artículo pretendemos mostrar porqué, a nuestro entender, la comunidad científica y en especial los que trabajamos en el ámbito de la criptografía y la seguridad de la información, debemos comprender el funcionamiento de la moneda digital Bitcoin. Como se verá, los motivos que presentamos trascienden a la propia moneda Bitcoin y se centran en la red peer-to-peer (P2P) subyacente a dicha moneda, que proporciona un sistema distribuido que permite mantener un registro público también distribuido. Dicho registro permite distintos usos y, como se verá, deja la puerta abierta a múltiples innovaciones.**

***Palabras clave*—Bitcoin, criptomoneda (*cryptocurrency*), P2P, Sistemas Distribuidos (*Distributed Systems*), Problema de los Generales Bizantinos (*Byzantine Generals Problem*)**

I. Introducción

Los sistemas distribuidos presentan un sinfín de propiedades que los hacen unos candidatos idóneos en distintos escenarios. Por ejemplo, son sistemas altamente escalables, que pueden ofrecer rendimientos muy elevados. Por otro lado, en cuanto a la seguridad se refiere, un sistema distribuido presenta la ventaja de eliminar el único punto crítico que supone un sistema centralizado, así como la supremacía que implica el control de dicho punto crítico.

Sin embargo, uno de los problemas también de seguridad asociado a los sistemas distribuidos es la naturaleza poco *controlable* de las entidades que participan en el sistema distribuido. Las entidades que lo forman tienen cierto grado de autonomía y, por lo tanto, su comportamiento puede ser alterado, ya sea a causa de fallos no deseados dentro de la propia entidad, como a causa de la existencia de entidades con intereses contrarios al resto del sistema. Uno de los problemas de seguridad asociados a los sistemas distribuidos es el conocido como los generales bizantinos.

El **problema de los generales bizantinos** [1] es un experimento mental creado para ilustrar el dilema de lograr un consenso entre un conjunto de entidades con un objetivo común cuando entre ellas pueden existir traidores, es decir, entidades con objetivos opuestos que intenten dinamitar el proceso. Además, se supone que las comunicaciones entre dichas entidades son limitadas e inseguras. El problema se presenta como una analogía con un escenario de guerra, donde un grupo de generales bizantinos se encuentran acampados con sus tropas alrededor de una ciudad enemiga que desean atacar. Después de observar el comportamiento del enemigo, los generales deben comunicar sus observaciones y ponerse de acuerdo en un plan de batalla común que permita atacar la ciudad y vencer. Para ello, los generales se comunican únicamente a través de mensajeros. Además, existe la posibilidad que algunos de los generales sean traidores y, por lo tanto, decidan enviar mensajes con información errónea con el objetivo de confundir a los generales leales. Un algoritmo que solucione el problema debe asegurar que todos los generales leales acuerdan un mismo plan de acción y que unos pocos traidores no pueden conseguir que el plan adoptado por los generales leales sea equivocado.

Uno de los grandes logros que supone Bitcoin, más allá de ser la primera criptomoneda con una aceptación extendida[1] por todo el mundo, es el hecho de ofrecer la primera solución práctica al problema de los generales bizantinos. La aplicación de los generales bizantinos a la criptomoneda permite, por primera vez en la historia, transferir propiedad digital a otro usuario de Internet, de manera que solo el propietario pueda hacerlo, únicamente el destinatario pueda recibirla, todo el mundo pueda validar la transferencia y esta sea reconocida por todos los participantes, todo ello realizado de manera totalmente distribuida.

En este artículo, expondremos porqué es interesante conocer la criptomoneda Bitcoin y repasaremos las aportaciones que el esquema utilizado por Bitcoin representan, más allá de la propia moneda.

El resto del artículo se estructura de la siguiente manera: la Sección II presenta a grandes rasgos el sistema Bitcoin; después, la Sección III enfatiza las características de Bitcoin en relación a la notarización de información; posteriormente, la Sección IV comenta extensiones de la notarización que se han propuesto, tanto como para el propio sistema Bitcoin como para sistemas posteriores construidos a su imagen; seguidamente, la Sección V menciona algunas de las aplicaciones que un sistema de notarización distribuido puede tener; finalmente, la Sección VI presenta las conclusiones.

II. Bitcoin: conceptos básicos

Dado que este artículo pretende resaltar las características que hacen del sistema Bitcoin un sistema a tener en cuenta

[1]Trabajos existentes realizados con datos de Enero de 2014 [2] descubren alrededor de 110000 nodos diferentes conectados en un día cualquiera.

ISBN: 978-84-9717-323-0

en distintos ámbitos más allá de la propia moneda, en esta sección se describen únicamente unas nociones muy básicas del funcionamiento de los Bitcoins, imprescindibles para que el lector comprenda el abasto de las contribuciones que Bitcoin representa.[2] Por este motivo, es posible que dicha descripción sea incluso insuficiente para entender la corrección y completitud del sistema Bitcoin como moneda digital. El lector interesado en conocer a fondo el funcionamiento de la moneda puede obtener más información en: [3], [4], [5].

II-A. Las transacciones

La unidad básica de funcionamiento de Bitcoin son las llamadas **transacciones**. Una transacción indica un movimiento de Bitcoins de una dirección de origen a una dirección de destino. Cada **dirección** de Bitcoins representa una clave pública (Bitcoin se basa en criptografía de curvas elípticas). Para **gastar** Bitcoins es necesario conocer la clave privada asociada a la clave pública que contenga un saldo en Bitcoins. Entonces, se pueden gastar esos Bitcoins, es decir, transferirlos a otra dirección, firmando digitalmente con la clave privada la transmisión de esta información y enviando la nueva transacción a toda la red. Veámoslo con un ejemplo:

Sea $\{PK_A, SK_A\}$ ($\{PK_B, SK_B\}$) el par de claves, pública y privada, del usuario Alice (respectivamente, del usuario Bob). La función $Addr(PK)$ nos devuelve la dirección de Bitcoin asociada a la clave pública PK, H es una función hash y $Sig_{SK}(m)$ representa la firma de m con la clave privada SK. Supongamos que Alice ha recibido anteriormente en una transacción T_0 la cantidad de $25BTC$ a su dirección, $Addr(PK_A)$:

$$\begin{aligned} T_0 &= \{input_0, output_0\} \\ input_0 &= \{\cdots\} \\ output_0 &= \{Addr(PK_A), 25\} \end{aligned}$$

Alice desea, entonces, enviar los $25BTC$ a Bob. Para ello, Alice crea una nueva transacción, T_1:

$$\begin{aligned} T_1 &= \{input_1, output_1\} \\ input_1 &= \{H(T_0), Sig_{SK_A}(T_0 + output_1), PK_A\} \\ output_1 &= \{Addr(PK_B), 25\} \end{aligned}$$

Veamos el motivo de incluir cada uno de los elementos en la transacción. En primer lugar, la transacción nueva T_1 incluye el hash de la transacción que se quiere gastar, T_0, que actúa como un puntero. En segundo lugar, Alice, que es la propietaria de la dirección que contiene los fondos, es la única que puede gastarlos ya que es la única que conoce la clave privada SK_A necesaria para realizar la firma $Sig_{SK_A}(T_0 + output_1)$. Además, si Alice no ha usado anteriormente esta dirección, ella es también la única que conoce su clave pública PK_A, ya que la función $Addr$ es pública pero no invertible. Por este motivo, para que se pueda validar la firma, la transacción debe incluir PK_A. Por último, Alice indica que quiere transferir los fondos a Bob firmando la dirección de Bob juntamente con el importe a transferir ($output_1$). De este modo, solamente Bob, que es el único conocedor de su clave privada, podrá gastar la transacción T_1.

Bob puede verificar que le han sido transferidos los fondos comprobando que $Addr(PK_A)$ coincida con la dirección de destino de T_0 y que la firma $Sig_{SK_A}(T_0{+}output_1)$ es correcta con PK_A.

II-B. La cadena de bloques

Tal como hemos descrito el sistema hasta este punto, no hay nada que impida a Alice gastar repetidamente los $25BTC$ que ha recibido en la transacción T_0, es decir, crear $T_1, \ldots, T_i$ transacciones con direcciones de destino diferentes utilizando la misma dirección de origen y el mismo puntero a la transacción anterior. Este comportamiento se conoce bajo el nombre de **doble gasto** y, obviamente, es necesario prevenirlo en cualquier tipo de moneda virtual.

Con el objetivo de prevenir el doble gasto, Bitcoin anota todas las transacciones ocurridas en un registro común conocido como **cadena de bloques** (o *blockchain*). De este modo, cuando Bob recibe la transacción T_1 de Alice, puede acudir al registro público y comprobar que Alice no haya gastado anteriormente el dinero que le está transfiriendo, es decir, comprobar que no existe ninguna otra transacción que tiene en su $input$ el mismo valor $H(T_0)$.

Este registro único se genera, distribuye y almacena de forma distribuida, de modo que todos los participantes están de acuerdo en su contenido sin la intervención de ninguna autoridad central. Es en esta creación de un registro público único de manera distribuida donde Bitcoin resuelve de manera práctica el problema de los generales bizantinos y por el cual el potencial de Bitcoin sobrepasa de largo el de una moneda virtual.

El registro público de Bitcoin (la cadena de bloques) está formado, como su nombre indica, por un conjunto de **bloques** enlazados de manera secuencial. Con el paso del tiempo, nuevos bloques son creados y añadidos a la cadena existente. La cadena de bloques es, por lo tanto, un registro que solo permite anexar información. Cada bloque contiene una cabecera y una carga útil. La carga útil son las transacciones que han ocurrido en el sistema desde que se creó el último bloque. De este modo, el conjunto de transacciones aceptadas como válidas por la red son las transacciones contenidas en cada uno de los bloques que pertenecen a la cadena de bloques. A su vez, la cabecera de cada bloque contiene un puntero al bloque anterior, de modo que los bloques forman una cadena. Además, la cabecera contiene también un valor de *nonce*, que permite crear bloques válidos como veremos a continuación.

Los usuarios que se dedican a crear bloques en la red Bitcoin son conocidos como **mineros**, y son una pieza fundamental del esquema. Cualquier usuario de la red puede ser un minero. Su trabajo consiste en validar las transacciones que se envían por la red P2P, incluyendo las válidas en nuevos bloques y

[2] De hecho, se presenta una simplificación del esquema que no corresponde exactamente al protocolo Bitcoin, pero que permite entender sus puntos clave sin entrar en todos los detalles.

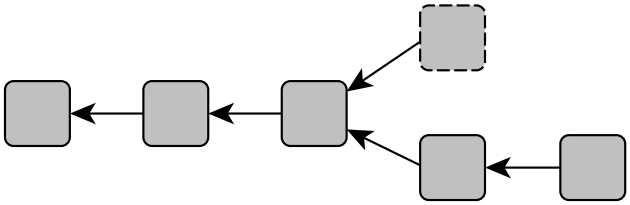

Figura 1. Bifurcación de la cadena.

descartando las inválidas. De este modo, si una transacción intenta gastar un importe ya gastado, o bien un usuario intenta gastar una transacción que no le pertenece (generando por lo tanto una firma inválida), esta nueva transacción nunca será incluida en un bloque y, de este modo, no habrá existido para el sistema. Por lo tanto, se necesita asegurar que los mineros hacen su trabajo correctamente, es decir, que aunque existan algunos mineros *traidores* que actúen en contra del interés común, se asegura que los mineros *leales* consigan acordar una cadena única, que contenga únicamente transacciones válidas.

Para lograrlo, se requiere que los bloques contengan una **prueba de trabajo** (*proof-of-work*) para ser considerados válidos. Dicha prueba de trabajo demuestra que el minero ha gastado un tiempo de computación en la generación del bloque. De este modo, mientras el poder de cómputo de la red esté distribuido, un grupo pequeño de mineros *traidores* no podrá modificar la cadena de bloques en su beneficio. La prueba de trabajo que utiliza Bitcoin consiste en encontrar un valor de *nonce* para el nuevo bloque de tal manera que el hash del bloque sea inferior a un valor objetivo fijado. Por las propiedades de las funciones hash, la única manera de conseguir un hash inferior al valor objetivo es ir probando diferentes valores de *nonce*, hasta dar con uno que genere el hash buscado.

Una vez un minero ha encontrado un bloque que cumple los requisitos, lo envía a toda la red, de manera que el nuevo bloque se convierte en el último de la cadena. A partir de ese momento, todos los mineros actualizan su estado, es decir, actualizan el puntero al último bloque conocido y actualizan las transacciones conocidas por el sistema, incluyendo en la generación del nuevo bloque solo aquellas que no se encuentran ya en la cadena.

Se puede dar el caso que dos mineros encuentren dos bloques distintos válidos que tengan el mismo bloque padre de manera más o menos simultánea (Figura 1), y que ambos envíen los bloques encontrados a toda la red. En este caso, se acepta el bloque que genere la cadena más larga, en términos del trabajo invertido en realizarla.

Como se ha visto, la existencia de mineros es fundamental para el funcionamiento del sistema, así que es necesario asegurar que existen **incentivos** suficientes para que los usuarios de la red quieran realizar el trabajo de minería, cosa que supone un coste (al menos en electricidad) para ellos. Actualmente el incentivo principal de los mineros es la recompensa que reciben, en forma de Bitcoins, cada vez que generan un bloque. Hemos visto como se transferían Bitcoins de una dirección a otra pero, hasta este momento, no hemos comentado como se crean estos Bitcoins. Los Bitcoins se crean a partir de un tipo de transacción especial, la **transacción de generación**, que se incluye en cada bloque. Dicha transacción tiene una dirección de destino (que pertenece al minero que se ha generado el bloque) pero no tiene ninguna dirección de origen. El importe de esta transacción de generación va disminuyendo con el tiempo y, a día de hoy, es de $25BTC$[3]. Cada bloque solo puede contener una única transacción de generación.

III. Notarización de información en el protocolo Bitcoin

En esta sección, describiremos las contribuciones de Bitcoin con relación a la notarización de información, es decir, a la creación de un registro único común de manera totalmente distribuida.

Bitcoin utiliza la cadena de bloques para almacenar transacciones, es decir, las unidades de información almacenadas en el registro único de Bitcoin son transacciones. Estas transacciones han sido creadas con anterioridad por algún miembro de la red, y difundidas por toda la red.

Suponiendo que existen usuarios en la red creando transacciones, el trabajo de los nodos de Bitcoin, es decir, de las entidades que forman parte del protocolo distribuido para crear el registro común de información, se resume en cuatro grandes tareas: validación, afianzamiento, transmisión y almacenaje.

III-A. Validación

Los mineros validan cada una de las transacciones que se incluyen en un bloque. Sea T_1 la transacción a validar, las comprobaciones a realizar son las siguientes:

- No existe doble gasto, es decir, T_1 no intenta gastar una transacción anterior T_0 ya gastada anteriormente.
- La transacción anterior T_0 que se intenta gastar existe.
- La clave pública especificada en la entrada de T_1 se corresponde a la dirección de salida especificada en T_0.
- La firma es correcta al validarla con la clave pública especificada en la entrada de T_1.

Aunque el funcionamiento de Bitcoin es muy similar al que hemos descrito, en realidad su especificación no se describe en estos términos sino en otros mucho más generales, con el objetivo de permitir realizar transacciones más complejas. En vez de fijar como se deben codificar las claves públicas, las direcciones y las firmas dentro de cada transacción, Bitcoin dispone de un **lenguaje de scripting** propio basado en pila, el código del cual se inserta tanto en las salidas como en las entradas de las transacciones. A la hora de validar una transacción, se apila el *script* de entrada con el de salida y se evalúa el *script* resultante. Si el resultado final de la evaluación es Cierto, entonces la transacción se considera válida. En caso contrario, la transacción se considera inválida.

[3]En el momento de escribir estas líneas, en Febrero de 2014, este importe equivale a unos 20,000 dólares.

Las validaciones descritas en este apartado forman parte de lo que sería una validación completa. Nótese que para realizar esta validación es necesario conocer la cadena de bloques entera, juntamente con todas las transacciones que contiene. Esto tiene un coste de espacio elevado. Además, recorrer la cadena en busca de las transacciones implicadas supone también un coste computacional elevado, que junto al coste en espacio, suponen un problema para dispositivos ligeros como móviles o incluso ordenadores limitados. Por este motivo, Bitcoin dispone del **Protocolo de Validación Simplificado** (SPV, del inglés *Simplified Payment Verification*), que permite a un usuario comprobar que ha recibido un pago utilizando significativamente menos recursos a costa de una reducción en la seguridad de la validación. Para la validación SPV, solo es necesario disponer de una copia de las cabeceras de los bloques de la cadena (que el cliente puede pedir a otro(s) nodo(s) de la red en cualquier momento) así como de algunos valores hash que permiten localizar la transacción dentro del bloque.

III-B. Afianzamiento

Además de validar las transacciones, es necesario también crear los bloques que las afianzan, así como validar a su vez la corrección de estos bloques. Este proceso es el que permite construir el registro común único y se realiza de manera totalmente distribuida en Bitcoin.

Como hemos visto, el afianzamiento en Bitcoin se basa en una prueba de trabajo (*proof-of-work*), consistente en encontrar un valor de *nonce* para el bloque B de tal manera que $H(B) < t$, es decir, que el hash del bloque sea inferior al objetivo fijado. El valor objetivo no es constante, y permite adaptar la dificultad de la prueba al poder de cómputo de la red en cada momento, con el propósito de generar un nuevo bloque cada 10 minutos.

De este modo, si un atacante quiere modificar la cadena de bloques, ya sea para dar marcha atrás y anular transacciones que ha realizado, ya sea para tener control de lo que se anota en el registro común, será necesario que éste disponga de un poder de cómputo superior al $50\,\%$ de la red[4]. En caso contrario, si el atacante intenta modificar la cadena de bloques generando sus propias alternativas, no tendrá suficiente poder de cómputo como para generar bloques más rápido que el resto de la red, por lo que su rama no será la más larga, y será descartada.

III-C. Transmisión

Bitcoin utiliza una red P2P totalmente distribuida para propagar la información. Bloques y transacciones son transmitidos a través de esta red.

Cuando un nodo quiere realizar una transacción (o bien encuentra un bloque válido), este lo envía a toda la red. Para hacerlo, lo envía a los nodos que se encuentran directamente conectados con él y éstos, a su vez, lo reenvían a sus vecinos, siempre que el objeto en cuestión (bloque o transacción) sea válido. De este modo, la información se propaga por toda la red.

Dado que, a diferencia de los bloques, las transacciones no contienen ninguna prueba de trabajo, un nodo malicioso podría crear un gran número de transacciones válidas con la intención de desbordar la red. Para evitar este tipo de ataques, los nodos estándar de Bitcoin aplican una política de retransmisión de transacciones, que obliga a incorporar una **comisión** a las transacciones que cumplen ciertas características que las hacen ideales para este tipo de ataques. Aún así, los usuarios que realizan transacciones tienen libertad para decidir si pagan o no una comisión y, en caso de hacerlo, del importe que esto conlleva. Estas comisiones afectan, como hemos comentado, la retransmisión de la transacción, además de su inclusión en un bloque. Esto último es debido a que el minero, además de cobrar la recompensa por encontrar un bloque, también obtiene todas las comisiones que las transacciones que contiene el bloque incorporan. Por este motivo, incluir comisiones en las transacciones puede crear incentivos adicionales para que los mineros las incluyan en sus bloques.

III-D. Almacenaje

El almacenaje de la cadena de bloques se lleva a cabo con mucha redundancia: todos los nodos completos de la red contienen una copia entera de la cadena de bloques (y sus transacciones). Esto permite a estos nodos validar de manera correcta cada nueva transacción.

Tener que mantener una copia completa de la cadena puede suponer un problema para los nodos operando en dispositivos ligeros como, por ejemplo, dispositivos móviles. En Febrero de 2014, después de 5 años de operación de la moneda Bitcoin, la cadena de bloques ocupa unos 13 GB.

IV. Extensiones para la notarización de la información

En esta sección, repasaremos algunas de las mejoras o alternativas que se han propuesto sobre el protocolo de Bitcoin, algunas de ellas implementadas ya en otras criptomonedas, otras solo presentadas a nivel teórico.

IV-A. Validación

Aunque los *scripts* de Bitcoin permiten especificar qué se necesita para poder gastar una transacción, el lenguaje es limitado. Según la propia descripción del lenguaje, este **no** es **Turing-completo** por diseño, argumentando motivos de seguridad para justificar esta decisión. Si bien es cierto que esto previene de realizar ciertos ataques (pensemos, por ejemplo, en un *script* con un bucle infinito, que se ejecutaría de manera indefinida cada vez que se intentara validar), también limita el conjunto de programas que se pueden codificar con él.

Una extensión que se ha propuesto en este sentido es incorporar un lenguaje Turing-completo a las transacciones [7], aumentando así la potencia de las mismas. Este lenguaje debe ir acompañado de un sistema de seguridad que permita evitar ciertos ataques, como el anteriormente comentado *script* de ejecución infinita. Una de las propuestas contempla incluir

[4]Estudios recientes presentan un ataque teórico que reduce este valor al $33\,\%$ del poder de cómputo total[6].

una comisión que se debe pagar por cada paso de ejecución del algoritmo, de manera que los *scripts* más simples, que suponen menos tiempo de validación, resulten más baratos que aquellos más complejos, que necesitan gastar tiempo de computación para ejecutarse.

Otra de las limitaciones del protocolo Bitcoin se encuentra en relación al Protocolo de Validación Simplificado. El protocolo se puede llevar a cabo para el tipo de transacción estándar dentro de Bitcoin, pero se complica enormemente (hasta el punto que no se ha encontrado solución aún) para ciertas variaciones del esquema.

IV-B. Afianzamiento

Bitcoin utiliza una prueba de trabajo basada en el cálculo de hashes para afianzar la información. A día de hoy[5], se estima que la red dispone de un poder de cómputo superior a los 23000 TH/s. Esto supone un gasto energético elevado, hecho que ha empezado a causar alarma por los posibles efectos negativos sobre el medio ambiente. Además, dicho gasto energético únicamente se utiliza para la propia criptomoneda ya que el cálculo de los hash para afianzar los bloques no tiene ningún otro fin. Por lo tanto, es interesante plantearse alternativas a la prueba de trabajo basada en hash que permitan obtener una funcionalidad equivalente. Se han propuesto cuatro enfoques diferentes:

Proof-of-Work: Como hemos visto, consiste en demostrar que se ha realizado una cantidad de trabajo para conseguir el bloque. Por lo tanto, la probabilidad de conseguir minar un bloque depende del poder de cómputo empleado en el trabajo. En este ámbito, las mejoras se centran en dos alternativas. Por un lado, proponer funciones que no requieran una inversión en hardware para el minado de bloques (como sucede actualmente con la función SHA256), para democratizar el proceso de minado y evitar así grandes clústers de minado que pudieran llegar a controlar la red. Dentro de esta alternativa se encuentran funciones hash, como por ejemplo scrypt [8] que requieren un volumen elevado de memoria para su cálculo, haciendo poco viable la creación de hardware específico. Otro enfoque, mucho más ambicioso, es la propuesta de una función de *proof-of-work* tal que su propio cálculo permita resolver problemas útiles computacionalmente costosos. El problema principal de este enfoque es formalizar problemas que tengan las siguientes propiedades, necesarias para una *proof-of-work* utilizada como sistema de validación de los bloques de la cadena: 1) verificabilidad: el problema propuesto debe ser difícil de realizar pero, una vez resuelto, la validación de la solución encontrada debe ser muy simple; 2) granularidad: la dificultad del problema propuesto debe ser granular, en el sentido que se debe permitir ajustar la dificultad del mismo de forma controlada y progresiva. En la actualidad únicamente se conoce una *proof-of-work* con estas características, utilizada en la moneda digital PrimeCoin [9]. En este caso, la *proof-of-work* consiste en encontrar ciertas cadenas de números primos, en concreto, cadenas de Cunningham de primera y segunda especie o cadenas de primos gemelos.

Proof-of-Stake: En este caso, la probabilidad que un minero encuentre un bloque depende de la cantidad de Bitcoins que posee actualmente. De este modo, mientras la posesión de Bitcoins sea distribuida, también lo será la capacidad de minar.

Proof-of-Burn: En este tipo de pruebas, la probabilidad de conseguir afianzar un bloque depende del número de Bitcoins destruidos expresamente para este propósito, es decir, quemados. Destruir Bitcoins es tan sencillo como enviarlos a direcciones que no se puedan gastar, es decir, a *scripts* que se evalúen a Falso de manera deliberada.

Proof-of-Excellence: En este sistema definido vagamente en [10], se crean torneos periódicamente y se minan bloques en función del rendimiento de cada participante en el torneo.

IV-C. Transmisión

Algunas criptomonedas surgidas después del auge de Bitcoin modifican el tiempo medio necesario para crear un bloque, fijado en 10 minutos en Bitcoin. Aunque parezca un cambio trivial, esto tiene consecuencias importantes sobre la seguridad del esquema.

Por un lado, la seguridad de una transacción en Bitcoin se mide utilizando el número de **confirmaciones** que ésta tiene, es decir, cuántos bloques se han añadido a la cadena después del bloque que contiene la transacción en cuestión. El motivo es que, como más confirmaciones tenga una transacción, más difícil es anularla, ya que para ello habría que construir una rama alternativa de la cadena que supere en dificultad a la rama actual. Bajo este punto de vista, fijar un tiempo de creación de bloques de 10 minutos hace de Bitcoin un sistema lento en dar por válidas las transacciones. El cliente estándar espera a que existan 6 confirmaciones antes de aceptar una transacción como pago, lo que fijaría un tiempo medio de 1 hora para el proceso.

Por otro lado, cuando un nuevo bloque es encontrado por un minero, este lo envía a sus vecinos, de modo que el bloque se propaga por la red. Esta propagación no es instantánea, y son necesarios algunos segundos para que los nodos la reciban[2]. Durante este tiempo de propagación, el minero que ha encontrado el bloque ya se encuentra minando encima de este, mientras que el resto de mineros aún trabajan en el bloque anterior. Esto tiene dos consecuencias importantes. La primera es que estos últimos mineros están realizando trabajo inútil. El porcentaje de trabajo inútil por bloque, suponiendo un tiempo de propagación constante, es mayor como menor sea el tiempo de generación de bloques. La segunda consecuencia se deriva también de este problema, ya que el minero que ha encontrado el bloque se encuentra en clara ventaja respecto al resto de la red. Esto también se acentúa con la disminución del tiempo de generación de los bloques.

Una de las propuestas para minimizar el impacto que el trabajo inútil sobre bloques ya minados supone para el sistema es la de recompensar no solo al bloque que queda en la cadena principal, sino también a algunos de los bloques válidos que hayan quedado en otras bifurcaciones de la cadena [7].

[5]Febrero 2014

IV-D. Almacenaje

En relación al almacenaje de la información, el principal problema que Bitcoin tiene que afrontar es la escalabilidad. Con la continua creación de nuevas transacciones, el tamaño de la cadena de bloques no hace más que aumentar a buen ritmo, augurando problemas de almacenamiento a largo plazo. Por otro lado, las restricciones en el tamaño de los bloques implican que en la actualidad la red bitcoin solamente pueda procesar un máximo de 7 transacciones por segundo[6], un valor demasiado pequeño para una moneda con vocación global.

Aunque de momento no se ha implementado ninguna solución, se discute activamente la posibilidad de incorporar un algoritmo de poda de la cadena, de manera que no sea necesario guardar todas las transacciones. Así, transacciones antiguas podrían ser eliminadas, guardando de ellas solo su hash, para preservar la integridad de la cadena.

V. Posibles aplicaciones

Las utilidades prácticas de Bitcoin (o de un sistema basado en la cadena de bloques) sobrepasan de largo las de una simple moneda. A continuación, se listan algunas de las aplicaciones que el sistema proporciona, tanto aquellas de las que ya existen implementaciones sobre Bitcoin, como aquellas que surgen a partir de alternativas derivadas, así como también las que de momento quedan en un plano teórico.

- Submonedas ([7], [11], [10], [12]): La cadena de bloques se puede utilizar para representar transacciones de otros bienes, como por ejemplo, otras monedas, oro, acciones o propiedad.
- Derivados financieros ([7]): Se pueden representar también en la cadena de bloques derivados financieros, explicitando sobre que bien concreto se deriva el precio.
- Servicios de marca de tiempo o *timestamps* ([13], [11]): Incluyendo el hash de un archivo en un bloque de la cadena, se puede demostrar la existencia del archivo en el momento de la creación del bloque.
- Servicio de nombres de dominio o *DNS* ([11]): La cadena se puede utilizar también para almacenar información de nombres de dominio de manera totalmente distribuida.
- Sistemas de Reputación Anónimos ([7]): Del mismo modo que se pueden registrar nombres de dominio en la cadena, ésta se puede utilizar para construir sistemas de reputación anónimos.
- Cómputo multipartito seguro o *Secure multiparty computation* ([14], [15]): Protocolos para el cómputo bipartito y multipartito seguro se han propuesto recientemente, e incluso se han realizado implementaciones de algunos de los protocolos sobre Bitcoin.
- Juegos de Azar P2P ([16], [7]): Juegos de azar o loterías pueden implementarse de manera que éstos resulten seguros para todas las partes, utilizando trozos de la cadena (o hashes de estos) como generadores pseudoaleatorios.

[6]El tamaño máximo de un bloque es de 1MB y el tiempo entre bloques es de 10 minutos. Esto proporciona 1,7KB por segundo, lo que suponen unas 7 transacciones de 250bytes.

VI. Conclusión

Bitcoin es la primera moneda criptográfica que ha tenido una grande aceptación entre la población, existiendo implementaciones del cliente estándar que permiten operar con ella para múltiples plataformas. Este hecho, por si solo, ya tiene un gran mérito. Además, más allá de ser una criptomoneda en utilización, con un esquema criptográfico robusto, totalmente descentralizada y anónima, Bitcoin resuelve de manera práctica el problema de los generales bizantinos, permitiendo crear un registro único común de manera descentralizada. Los usos de este registro sobrepasan de largo los de la propia criptomoneda y, en consecuencia, creemos que es importante dar a conocer su existencia. Como hemos expuesto, ya existen diferentes iniciativas que hacen uso de este registro con finalidades muy diversas y, a nuestro parecer, estas iniciativas son solo el principio de una larga lista de aplicaciones que se pueden diseñar e implementar en base a este registro.

Agradecimientos

Este trabajo está parcialmente financiado por el Ministerio de Educación, a través de los proyectos TIN2011-27076-C03-02 CO-PRIVACY, TIN2010-15764 N-KHRONOUS, CONSOLIDER INGENIO 2010 CSD2007-0004 ARES, y de la beca FPU-AP2010-0078.

Referencias

[1] Lamport, Shostak, and Pease, "The Byzantine Generals Problem," in *Advances in Ultra-Dependable Distributed Systems, IEEE Computer Society Press*, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.1697

[2] J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí, "The Bitcoin P2P Network," in *Proceedings of the 1st Workshop on Bitcoin Research (in Assocation with FC14)*, ser. to appear, 2014.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] M. Nielsen, "How the Bitcoin protocol actually works," 2013. [Online]. Available: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

[5] Bitcoin community, "Bitcoin wiki." [Online]. Available: https://en.bitcoin.it

[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proceedings of the 1st Workshop on Bitcoin Research In Assocation with Financial Crypto*, 2014.

[7] V. Buterin, A. di Lorio, C. Hoskinson, and M. Alisie, "Ethereum white paper," 2013. [Online]. Available: http://www.ethereum.org

[8] C. Percival, "Stronger key derivation via sequential memory-hard functions," 2012.

[9] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," 2013. [Online]. Available: http://primecoin.io/bin/primecoin-paper.pdf

[10] S. King and S. Nadal, "Ppcoin," 2012. [Online]. Available: http://peercoin.net/bin/peercoin-paper.pdf

[11] vinced, "Namecoin: a secure general purpose p2p key/value storage system." [Online]. Available: http://namecoin.info/

[12] Bitcoin Wiki, "List of alternative cryptocurrencies," Última consulta: febrero 2014. [Online]. Available: https://en.bitcoin.it/wiki/List_of_alternative_cryptocurrencies

[13] Shesek, "BTProof: trusted timestamping on the bitcoin blockchain," 2013. [Online]. Available: https://www.btproof.com/

[14] A. M., D. S., M. D., and M. L., "Fair two-party computations via bitcoin deposits," in *Proceedings of the 1st Workshop on Bitcoin Research In Assocation with FC*, 2014.

[15] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in *to appear*, 2014.

[16] E. Voorhees, "Satoshidice," 2012. [Online]. Available: https://en.bitcoin.it/wiki/SatoshiDice